

# COMP3111 - Software Engineering

## T-33 Documentation

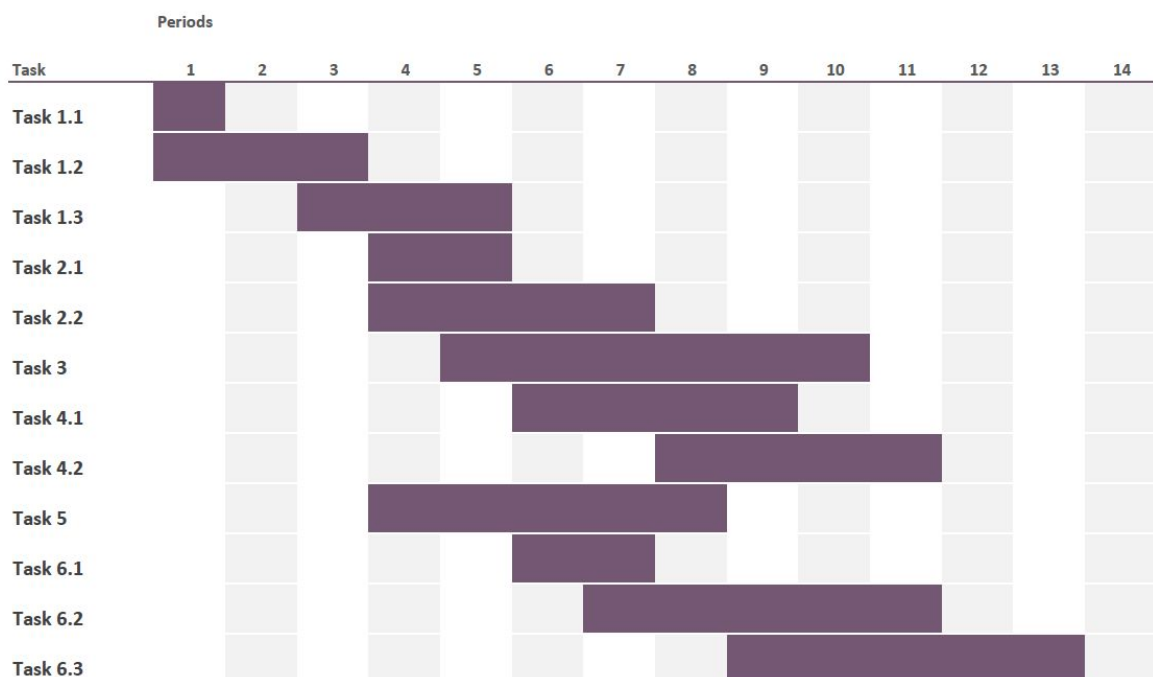
### Table of Contents

Gantt Chart	1
Burn Down Charts	2
Minutes of the 1st Project Scrum Meeting	4
Minutes of the 2nd Project Scrum Meeting	5
Minutes of the 3rd Project Scrum Meeting	6

### Gantt Chart

Gantt Chart

## Project Gantt Chart

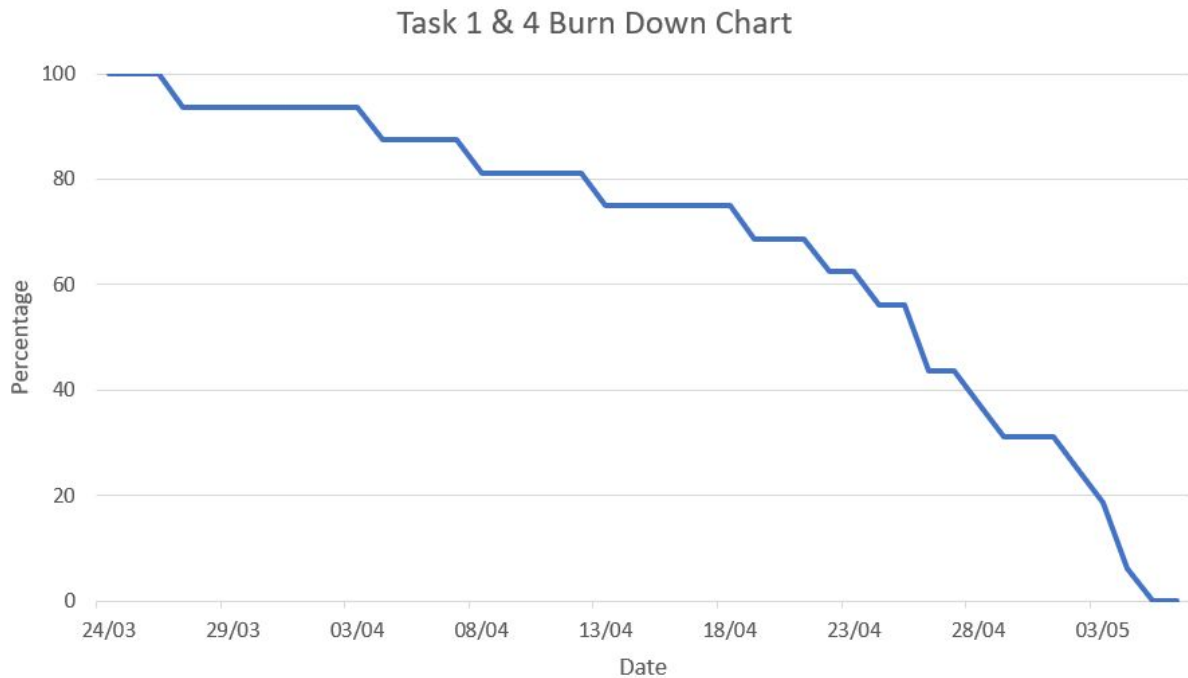


# COMP3111 - Software Engineering

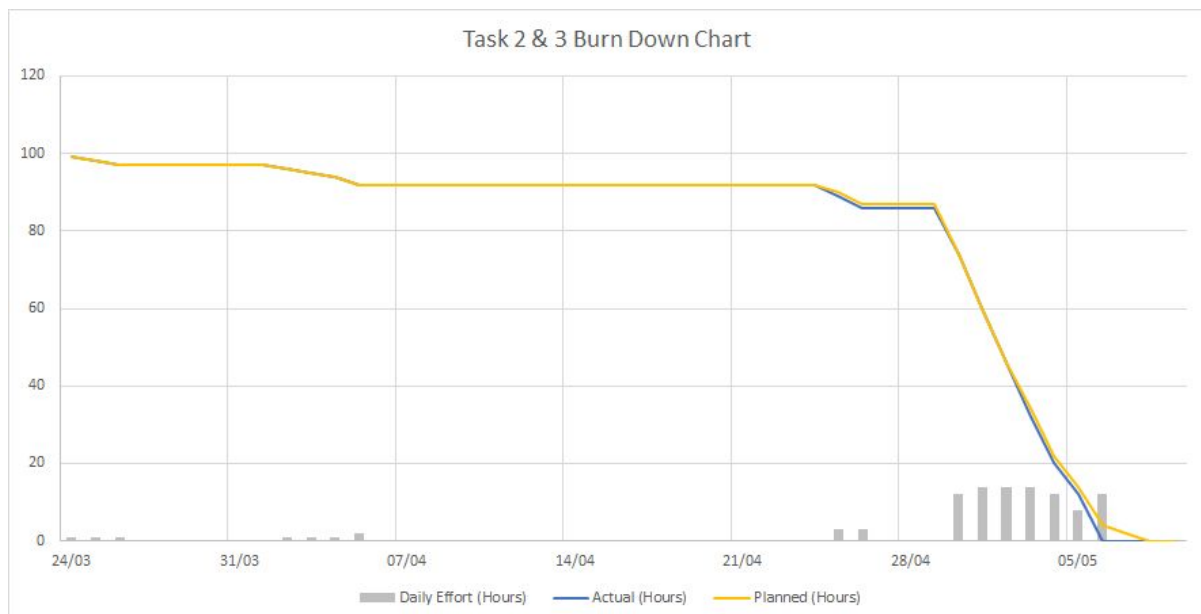
## Burn Down Charts

### Individual Burn Down Charts

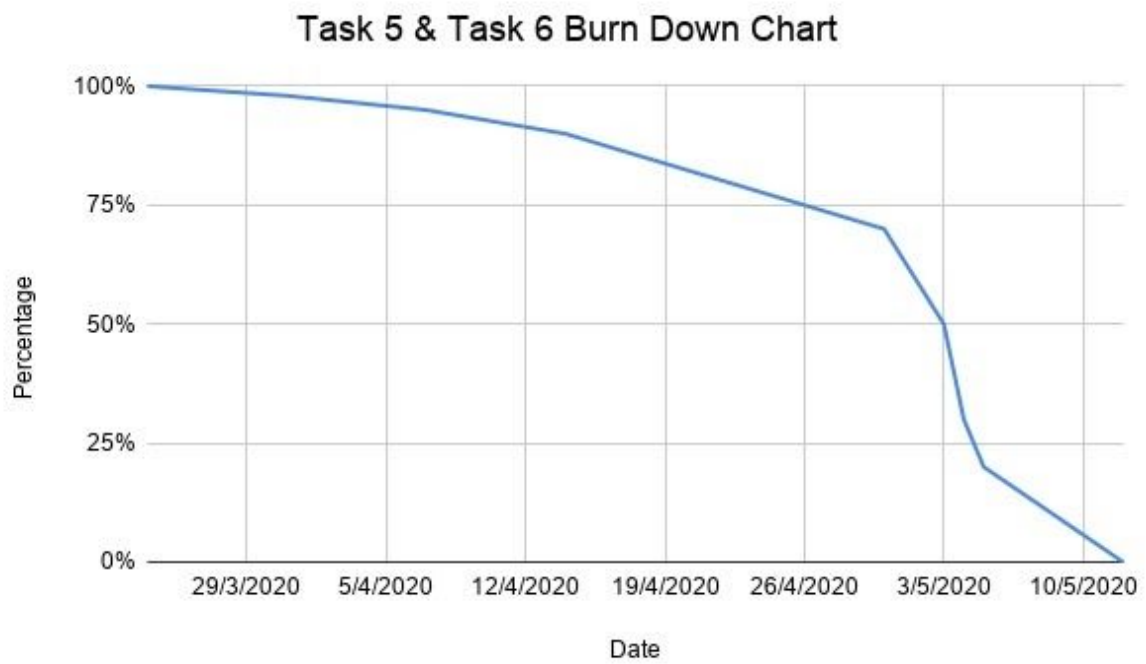
Task 1 & Task 4 - Pang Pak Long



Task 2 & Task 3 - Kong Ming Hin



## Task 5 & Task 6 - Ho Kin Chun



# **COMP3111 - Software Engineering**

## **Minutes of the 1st Project Scrum Meeting**

### **Progress**

- Understand the project description, familiar with our own tasks
- Understand the structure of the skeleton code, familiar with the user interface and the controller class.
- Understand the structure of scraper and how it captures the HTML elements.

### **Discussion**

- Implement a course, section, slot structure, where course class store a list of sections and section class store a list of slots
- Course title, credit, exclusion and description are stored in the course object
- Section code, section ID and list of instructors are stored in the section object
- Date and time are stored in the slot object

### **Goals to achieve**

- Draw class diagrams
- Draw use case diagrams
- Complete our own use case specification
- Implement the section class first
- Create skeletons that handle our own tasks

# **COMP3111 - Software Engineering**

## **Minutes of the 2nd Project Scrum Meeting**

### **Progress**

- Completed the skeletons for the system.
- Completed core classes that we need in the systems (Course, Section, Slot)

### **Discussion**

- 2019 winter, spring and summer term has a slightly different structure compared to 2019 fall, where the scraper is incapable to scrape these terms.
- Filter and enrolment list are implemented in separate classes, a list of enrolled courses is set up to keep track of the information in the controller class.
- The timetable function utilizes the enrolled course list and the function is called whenever any course is enrolled or dropped.
- Use a helper class for handling the filtering function instead of doing it inside the controller.
- Create a data model for entries inside the List tab, which stores the course code, sections, course name, instructors and enrollment information.
- Bind the UI and UI elements to the data model. Make sure the UI reflects the content of the data model correctly

### **Goals to achieve**

- Complete all the remaining tasks
- Update scraper to add support for winter, summer and spring term
- Make sure the system is working fine when combining multiple completed tasks together
- Bug fixes and code refactoring

# **COMP3111 - Software Engineering**

## **Minutes of the 3rd Project Scrum Meeting**

### **Progress**

- Completed tasks 1, 2, 3 and 4, including the followings:
- The scraper is capable of scraping all the term.
- The timetable is able to show enrolled course and time clashes, smaller font size is used to fit course title and section code into two lines.
- Fixed some bugs that occur after combining all completed tasks
- Fixed some visual bug that was overlooked when completing task 3
- Entries inside the List tab does not contain duplicated items
- All entries inside the List tab can now reflect the data model correctly

### **Discussion**

- Review the code to make sure no system requirements are missing (not captured)
- Refactor some of the code that is hard to maintain, hard to read and doing duplicated things
- Rework the filter class to make it fits the system requirements
- Add Javadoc
- Remove some unused functions

### **Goals to achieve**

- Make sure system requirements are captured completely and correctly
- Properly document all the classes and functions in the format of Javadoc
- Implement unit test
- Try to reach branch coverage as much as possible